(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 April 2001 (12.04.2001)

PCT

(10) International Publication Number
WO 01/26375 A1

(51) International Patent Classification⁷: H04N 7/24, 5/00

(21) International Application Number: PCT/EP00/09919

(22) International Filing Date: 6 October 2000 (06.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

99/12481	7 October 1999 (07.10.1999)	FR
00400941.1	5 April 2000 (05.04.2000)	EP
00402115.0	24 July 2000 (24.07.2000)	EP

(71) Applicant *(for all designated States except US)*: **THOMSON MULTIMEDIA** [FR/FR]; 46, quai Alphonse Le Gallo, F-92100 Boulogne-Billancourt (FR).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ABELARD, Franck** [FR/FR]; Thomson multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR). **DESCHAMPS, Fabien** [FR/FR]; Thomson multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR). **RABU, Christophe** [FR/FR]; Thomson multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR). **MAETZ, Pascal** [FR/FR]; Thomson multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR).

(74) Agent: **KOHRS, Martin**; Thomson multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR).

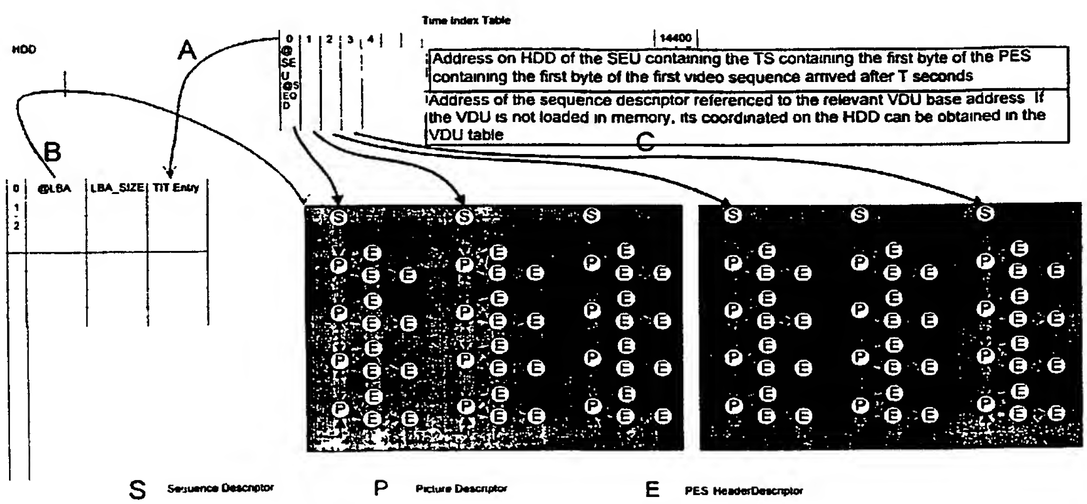
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND DEVICE FOR TRICKMODE GENERATION IN A DIGITAL VIDEO SYSTEM



(57) Abstract: The invention concerns a method for generating trickmode information for a digital video stream to be recorded in a digital video system. The method comprises the steps of: analyzing the structure of said video stream and deriving a plurality of descriptors of objects of the stream; writing said stream to a recording medium; inserting, into the object descriptors, information describing the address of the objects on the recording medium. Applicable in a digital television system in which receivers are fitted with recording means.

WO 01/26375 A1

Method and device for trickmode generation in a digital video system

The invention concerns a method for coding trickmode information in a digital video environment, and a device for generating this information.

An MPEG II or DVB compliant digital television stream comprises several layers, among which the elementary stream layer, the Packetized Elementary Stream (PES) layer and the Transport Stream (TS) layer. A corresponding decoder usually comprises a demultiplexer for filtering certain TS layer packets, a PES Parser for removing the PES layer and transferring the original elementary streams and at least a video decoder for decoding the video elementary stream.

Future decoders will incorporate mass storage devices in order to record compressed TS or PES streams. In order to implement trickmodes, such as slow or fast forward or backward play, the video stream needs to be edited before being transferred from the mass storage device to the video decoder. In particular for fast forward or backward play, only specific pictures or picture sequences are to be displayed. Thus, an efficient way of accessing this information on the recording medium is required.

An object of the invention is a method for generating trickmode information for a digital video stream to be recorded in a digital video system, characterized by the steps of:

analyzing the structure of said video stream and deriving a plurality of descriptors of objects of the stream;

writing said stream to a recording medium;

inserting, into the object descriptors, information describing the address of the objects on the recording medium.

Other characteristics and advantages of the invention will appear through the description of particular non-limiting embodiments of the invention, illustrated by the drawings among which:

- figure 1 is a block diagram of a television receiver according to the present embodiment,

- figure 2 is a diagram of the file system of a hard disk drive used as a mass storage medium according to the present embodiment,

- figure 3 is a diagram of the part of the file system dedicated to the recording and reproduction of audio/video streams,

- figure 4a is a diagram of an elementary storage unit ('SEU') used to store stream data, in PES mode, while figure 4b is a diagram of a SEU used to store stream data in Transport Stream mode,

- figures 5a and 5b are diagrams of the FIFOs used to store PES data to be written to the hard disk drive when in PES mode,

- figure 6 is a diagram representing different data structures for storing trickmode information according to the present embodiment,

- figure 7a is a representation of a PES layer video stream before insertion of a dummy PES header,

- figure 7b is a representation of the PES layer video stream of figure 7a after insertion of a dummy PES header,

- figure 8a is a representation of a TS layer video stream before insertion of a TS packet including a dummy PES header,

- figure 8b is a representation of a TS layer video stream of figure 8a after insertion of a TS packet including a dummy PES header,

- figure 9 is a representation of an elementary video stream seen by the input buffer of a video decoder.

The present description is made in the frame of a system accepting an MPEG II compliant data stream and uses the corresponding vocabulary. More information concerning the MPEG II standard syntax for video and transport level coding can be found for instance in the documents: ISO/IEC 13818-1 (Information Technology – Generic coding of moving pictures and associated audio information : Systems) and ISO/IEC 13818 – 2 (Information Technology – Generic coding of moving pictures and associated audio information : Video). The present system also complies with the DVB ETR-154 standard.

The invention is of course not limited to the MPEG II environment, or to the data layers described in the present patent application.

1. System overview

To achieve high quality trickmode management when playing back a video stream from a local mass storage device, the knowledge of the structure of the recorded video stream is required. This structure will be called trickmode information in what follows. It results from a parsing process carried out before and during the recording of the video stream. Parsing consists in analyzing the stream structure and in memorizing the nature of certain syntactical structures. Information relating to the structures, as well as their position on the mass storage medium, are also recorded.

According to the present embodiment, data such as but not limited to video is recorded at the transport stream layer or the packetized elementary stream layer. Trickmode information describes the structure of the stored video stream at a number of layers (Transport Stream (TS) – Packetized Elementary Stream (PES) – Elementary Stream (ES) according to the well-known MPEG II syntax), down to the compressed video information.

The main embodiment, carrying out recording at the TS layer level, will be described first. Differences with the second embodiment, which records at the PES layer level, will be indicated in each case. Both embodiments being compatible in the sense that both recording levels can cohabit in a same decoder, they will both be described in reference to figure 1.

(a) TS layer recording

Figure 1 is a block diagram of a digital television receiver according to the present embodiment. The receiver 1 comprises front-end circuitry 2 which can output a transport stream to a transport stream demultiplexer and filter 4. The front-end circuitry typically includes a tuner, an analog/digital converter, an appropriate demodulator and forward error correction circuits. It receives a signal from a signal source (not shown), which is typically a cable, a satellite dish and associated low-noise block and down-converter, or a terrestrial antenna. Global resources in the system comprise a RAM 5, a PES Parser 6, a second Transport Stream demultiplexer 7 , audio and video decoders 8 and 9 and a microprocessor 10. The TS filter and demultiplexer 4 is programmed by the microprocessor to filter and extract from incoming transport stream data packets corresponding to certain criteria, typically data packets having certain Packet Identifier (PID) values. Incoming stream content, in particular PID

assignment, is known for example from a certain number of transmitted data tables defined by the MPEG II standard or the DVB Service Information standard (Document reference ETSI EN 300 468). Private PID values may also be defined.

Filtered transport stream data packets are buffered in memory 5, a part of which is arranged as a TS Write FIFO 15, for further processing by Stream Parser 3.

Contrary to a conventional demultiplexer, which dispatches the different TS packets to separate buffers according to their PID value and thus their destination application (e.g. the audio and video decoders), the TS filter and demultiplexer 4 writes all packets corresponding to PIDs of streams to be recorded to a single buffer (i.e. TS Write FIFO 15 in the present embodiment), in the order of packet reception.

Compressed stream data and other data (e.g. control data) are transmitted between peripheral blocks through data paths modelised by the bus 11. The receiver further comprises a mass storage device 12, which according to the present embodiment is a hard disk drive. Mass storage device 12 is connected to bus 11 through an interface 13, in the present case an EIDE interface. The video decoder circuit 9 is connected in a known fashion to video processing and display circuitry 14.

Memory 5 contains the following areas:

- the already mentioned write FIFO 15 for storing filtered TS packet data to be written to the hard disk,

- a TS read FIFO 16 for storing TS packets data read from the hard disk,

- a trickmode buffer area 17 to store trickmode information to be written to, (or read from,) the hard disk.

(b) PES layer recording

For the purpose of PES layer recording, the memory 5 contains three write FIFOs referenced 18 to 20, respectively dedicated to Audio PES, Video PES and other data, and three read FIFOs referenced 21 to 23, also respectively dedicated to similar types of packets.

When the decoder functions in PES mode, the second demultiplexer 7 is not used, the PES packets being transferred directly from the hard disk 12 to the PES parser 6 through FIFOs 21 and 22.

The FIFOs 15, 16 and 18 to 23 are preferably organized in a circular manner.

2. Mass storage device

The hard disk drive file system will now be described. The disk drive 12 possesses a file system shown by the diagram of figure 2, the file system being dedicated to audio/video stream recording and reproduction. The file system responds to the specific requirements of the type of data which it manages. The present file system is optimized for sequential access of isochronous data streams, with blocks of relatively large size.

As a variant, a second file system (not illustrated) dedicated to the recording and retrieval of other data than streamed data may be present on the same hard disk. This second file system is optimized for random access to more conventional computer-type files. The boot block can be common to both file systems. This second file system is of a conventional type, such as a UNIX or MINIX file system, and will not be described in more detail.

Figure 3 takes a more detailed look at the stream file system. This file system comprises a superblock, a node storage area, a run extension storage area, an audio/video data storage area and a bit table area, which holds three bit tables describing the state of each elementary storage structure in each of the three storage areas.

The boot block comprises general information concerning the hard disk drive, such as volume name and volume identifier, BIOS parameters and a boot program.

The superblock contains information concerning the stream file system, in particular the addresses (under the form of logical block addresses – 'LBAs') and sizes of the different areas of the file system.

The node storage area is used to store nodes. A node is a data structure describing a file stored in the audio/video data storage area. It may also describe a directory. It contains such information as the file name, parent directory information and a description of the parts of the audio/video data storage area where the file is located. This information is given under the form of LBA runs, defined by an LBA starting address and a number of LBA blocks

forming the run. Since a limited number of runs may be stored in a given node, a pointer within the node may point to a run extension data structure located in the corresponding storage area. File location information is replaced by file or directory identifiers if the node is used to describe a directory. The first node describes the root directory.

The run extension storage area contains particular data structures identifying further LBA runs for a given file.

The bit table area contains three bit tables: the node bit table, the run extension bit table and the Storage Elementary Unit bit table. The first two tables respectively indicate the free or used state of each node, respectively run extension. The third table does the same for each elementary storage unit, which according to the present embodiment, represents a block of 128 Kbytes (of course, blocks of a different size and especially of larger size may be used, the 128 K value being given only as an example).

Finally, the audio/video data storage area comprises a series of elementary storage units ('SEU'). Each SEU comprises 256 sectors, thus representing 128 Kbytes.

Using the above data structures, the microprocessor 10 can create and delete files as well as write data to and read data from these files.

(a) For TS layer recording :

Figure 4b is a diagram of a SEU when it is used for TS layer recording.

The SEU comprises a short header and a payload made of a number of multiplexed whole TS packets. Since the SEU size is a multiple of 512 bytes since it contains an integer number of TS packets, a certain number of stuffing bits have to be added to the payload.

(b) For PES layer recording :

Figure 4a illustrates the contents of a PES stream SEU. The SEU comprises a header and, according to the present embodiment, up to three areas of varying size, respectively dedicated to video PES packets, audio PES packets and other PES packets.

The number of areas is not limited to three, although this is a realistic example. Several video elementary streams, audio elementary streams and auxiliary data streams may lead to a corresponding number of areas within a SEU. In this case, the memory 5 will contain a corresponding number of read/write FIFOs.

3. Recording process

(a) TS layer recording

The constitution of a SEU for TS layer recording and reproduction can best be explained by describing how filtered TS packets are handled by the different elements of the receiver. Once the demultiplexer has selected the packets corresponding to the programmed PID values, it stores them in the circular write FIFO 15 in memory 5. The type of content of a packet, i.e. video (V), audio (A) or other (O), is determined by the microprocessor 10 from the respective PID values in the packet headers. The content of video (V) transport stream packets processed by the demultiplexer is parsed, i.e. analyzed by the Stream Parser 6, for extraction of certain types of trickmode information described in more detail later. In principle, no such analysis is performed for audio or other data packets. The initial order of the TS packets in the stream is maintained in the FIFO 15. This way, the continuity counter values in the different packets remain coherent. Moreover, the synchronization between the different streams (in particular the video and audio streams corresponding to a same event) is maintained. Microprocessor 10 manages a read and a write pointer for the write FIFO 15. When the difference between the write and read pointers reaches the equivalent of 128 Kbytes minus the size of a SEU header, the microprocessor launches a write process to the hard disk.

For TS recording, each SEU header contains an indication of the length of useful data in the TS packet payload, in order to distinguish between TS packets and stuffing bits.

(b) PES layer recording:

In this case, the demultiplexer and filter 4 does not only filter TS packets: it also strips the TS layer away, before writing the TS payload, i.e. the PES packets, into RAM 5. PES packets are transferred to one of the circular

write FIFOs 18 to 20 depending on the value of the PID of the TS packet in which they were transported. Microprocessor 10 manages read and write pointers for each of these FIFOs. When the sum of all the differences between the write and read pointers for all buffers reaches the equivalent of 128 Kbytes minus the size of a SEU header, the microprocessor launches a write process to the hard disk. Video PES are parsed by Stream Parser 3 for trick mode information.

For PES recording, the header contains an information of the quantity of data of each type which is going to be written to the SEU, i.e. the size of each area associated with a specific PID, and the offset address of each area within the SEU. No stuffing bits are used in case of PES recording: PES packets may begin in one SEU and end in the following SEU.

The write process, be it for TS or PES recording, is started by the microprocessor 10, by sending an appropriate command to the EIDE interface, specifying the LBA address where the writing should start and the number of LBAs to be written. Once the hard disk drive is ready to carry out the writing process, the EIDE interface informs the microprocessor by an appropriate interrupt.

The write process continues by writing the SEU header content, generated by the microprocessor 10, to the HDD interface. The write process further continues by initiating DMA processes to the HDD interface 13 either from TS Write FIFO 15 (for TS recording) or in turn for each of the write FIFOs 18 to 20 (for PES recording). In a known fashion, HDD interface 13 comprises a cache memory acting as buffer for disk accesses.

It is of course supposed here that the proper file has been opened by the microprocessor and that the microprocessor has also indicated the destination SEU for the transferred data to the EIDE interface.

While this hard disk write process is taking place, packets (whether TS or PES) continue to be written to the FIFOs.

For PES recording, if figure 5a illustrates the FIFO and read and write pointer states just before transfer to the disk begins, then figure 5b represents the state once the transfer is achieved. When the pointers reach the top addresses of the FIFOs, they wrap around to the bottom addresses.

Although the FIFOs all have the same apparent size in figures 5a and 5b, different sizes may be used. A similar process applies for TS recording.

4. Trickmode data generation

Figure 6 is a diagram of the data structures used to store trickmode information. These structures and their storage will be discussed first, followed by the method for obtaining the corresponding data during stream recording.

A trickmode information data structure has several functions:

- It describes the video MPEG structure of the stream;
- It provides the necessary data for accessing the MPEG Video Access Units on the Hard Disk Drive and for transmitting them to the decoder;
- It allows random access to MPEG Video Access Units based on time indexing;
- This is a structure that can be used bi-directionally (i.e. in the forward and backward directions): descriptors of MPEG syntax elements are linked with each other according to their order in the stream, and the descriptors and tables are defined in such a way that it is easy to find a descriptor preceding or following a current descriptor;
- The trickmode data is spread over three different structures: a Video Description Unit Table (VDU Table), a Temporal Indexing Table (TT) and a number of blocks (Video Description Units – VDUs), for easy and fast access to information, depending on the circumstances;
- A Video Description Unit is re-locatable in memory. Relative address pointers are implemented for that purpose. Consequently, the trickmode information data structure, when stored on the Hard Disk Drive, can be loaded into memory by parts as needed, depending on available memory;
- The VDU Table contains information to access a VDU on the Hard Disk and to store it –partially or not- in memory.

Figure 6 shows two VDUs, appearing in gray. A VDU contains descriptors of a number of sequences, and for each sequence, descriptors relating to the PES Headers and the Pictures comprised in that sequence. As an example, in figure 6, each VDU contains three sequence descriptors, corresponding approximately to 1.5 seconds of video. Trickmode information is spread over a plurality of VDUs in order to enable the system to access only the

VDUs as necessary and to reduce the total amount of memory required to manipulate this data.

VDUs may also simply have a maximum size and describe a variable number of sequences. The knowledge of the maximum size of a VDU by the system allows to forecast memory needs. For example, during playback, it may be advisable to always have two VDUs in memory: the VDU which is currently processed, and the consecutive VDU.

Preferably, each VDU holds descriptors concerning entire sequences, not partial sequences. This avoids having to work on different VDUs (of which some may have to be loaded first) when working on the pictures of a same sequence.

The tables and explanations given below refer to the PES layer recording mode. For TS layer recording, an address of an item on the disk or in a SEU is to be replaced by the address of the TS packet header of the TS packet containing the first byte of the item.

According to the present embodiment, only entire SEUs are read from and written to the disk. Therefore, each stream object descriptor contains information which enables (a) transfer of corresponding SEUs from the disk drive, and (b) transfer from memory to the decoder.

For the step (a) (integer SEU transfer), the following data is used:
(1) Address of first SEU (Logical Block Address) to be transferred
(2) Number of SEUs to be transferred

For the step (b)
(3) Offset of object in the first SEU
(4) Length of object (for example in bytes)

Additional data, in particular data for linking the descriptors, is also provided in each descriptor, as indicated below.

Each sequence descriptor ("S" descriptor) comprises the data shown in table 1:

1	Sequence Index
2	PES Alignment
3	SEQ & EXT Headers Size in bytes unit
4	SEQ & EXT Headers Size in SEU unit
5	Pointer to the Descriptor of First Picture
6	Pointer to the Descriptor of the Last Picture.
7	Pointer to the Descriptor of next Sequence
8	Pointer to the Descriptor of previous Sequence
9	Logical Block Address of the SEU containing the first byte of the Sequence
10	Number of SEUs containing the whole sequence
11	Offset, in bytes, of the first byte of the sequence inside the first SEU containing this byte
12	Size of the sequence in bytes

Table 1 – Sequence descriptor

According to the present embodiment, a 'sequence' is an MPEG II sequence as defined in the ISO/IEC 13818-1 document.

The **sequence** index gives the rank of a sequence, compared to the beginning of the recorded video stream. It also plays the role of an identifier of the sequence descriptor.

The **PES Alignment** is a flag indicating whether PES headers in the sequence are immediately followed by a picture header or not.

The "**SEQ & EXT Headers Size in bytes unit** " is the size in number of bytes of the MPEG Sequence header plus all subsequent MPEG headers and extensions preceding the first Picture Header in the sequence.

This information enables a quick transfer of these headers (containing, among other data, the MPEG quantification tables, picture size...) to the decoder, or to transmit the sequence header along with the first picture of a sequence.

The "**SEQ & EXT Headers Size in SEU unit** " is the number of SEUs that contain a part of the MPEG headers and extensions mentioned in conjunction with the previous item, since these headers may be spread over several SEUs. In the present embodiment, this number is equal to two, at a maximum.

The "**Pointer to the Descriptor of First Picture** " is a pointer, in the VDU, to the Picture descriptor of the first picture in the considered sequence. Note that the sequence descriptor and the descriptors of all the pictures belonging to this sequence are, according to the present embodiment, by construction, always in the same VDU.

All pointers in the VDUs use a relative addressing scheme based on the VDU base address. This way, the VDU can be loaded into any memory area while retaining valid pointer values.

The "**Pointer to the Descriptor of Last Picture** " is a pointer, in the VDU, to the Picture descriptor of the last picture in the considered sequence.

The pointers to the first and last picture are useful for carrying out playback on forward, respectively reverse direction. As will be seen in conjunction with Table 2, each picture descriptor points to the previous and the next picture.

The "**Pointer to the Descriptor of the Next Sequence** " is a pointer, in the VDU, to the descriptor of the following Sequence in the stream. If this descriptor is not in the same VDU, then the pointer is null. In this case, the descriptor of the following sequence is the first sequence descriptor in the next VDU. This further VDU can easily be accessed with the use of the VDU Table.

The "**Pointer to the Descriptor of the Previous Sequence** " is a pointer, in the VDU, to the descriptor of the previous Sequence in the stream. If this descriptor is not in the same VDU, then the pointer is null. In this case, the descriptor of the previous sequence is the last sequence descriptor in the previous VDU. This previous VDU can be accessed with the use of the VDU Table.

The **Logical Block Address** (item 9) is the address of the SEU containing the first byte of the Sequence Header. In conjunction with item 11 (**Offset, from the SEU start, of the Sequence Header)**, the SEU read from the disk is easily parsed. The **SEU (Storage Elementary Unit) Address** is the address (Logical Block Address number), on the hard disk, of the 128 Kb block containing the first byte of the sequence header.

The Number of SEUs (item 10) indicates the number of SEUs which have to be loaded in order to have the whole sequence in memory. Indeed, according to one reading mode, at least one entire sequence is loaded at a time into memory, and images extracted therefrom one by one are then sent to the appropriate decoders. This kind of process is very efficient in reverse playback mode, as far as disk access efficiency is concerned.

Each Picture descriptor ("P" descriptor) holds the following data items:

1	Picture Index
2	Type
3	Temporal Reference
4	Field/Frame
5	Pointer to the descriptor of the next picture in the same sequence
6	Pointer to the descriptor of the previous picture in the same sequence
7	Pointer to the descriptor of the **aligned** PES packet **containing** the picture data.
8	Pointer to the descriptor of the first PES packet **starting inside** the current picture
9	First SEU Address
10	Number of SEUs containing the whole picture
11	First Byte Address in SEU: Offset, in bytes, of the first byte of the picture header inside the first SEU containing this byte
12	First Byte Address in Sequence: Offset, in bytes, between the first byte of the picture header and the first byte of the sequence containing this picture.
13	Size of the picture in bytes

Table 2 – Picture descriptor

The **picture index** gives the rank of a picture, compared to the beginning of the recorded video stream. It also plays the role of an identifier of the picture descriptor, in particular in relation with the Temporal Table described below.

The picture **Type** information indicates whether the picture is of the Intra, Predictive or Bi-directional coding type.

The **Temporal Reference** information is directly extracted from the MPEG II picture header. It gives the display order of the pictures relative to each other..

The **Field/Frame** information indicates whether the picture comprises an even field, an odd field or a whole frame.

The "**Pointer to the Descriptor of Next Picture** " is a pointer, in the VDU, to the Picture descriptor of the next picture in the current sequence. If the current picture is the last one in the sequence, then the pointer is null.

The "**Pointer to the Descriptor of Previous Picture** " is a pointer, in the VDU, to the Picture descriptor of the previous picture in the current sequence. If the current picture is the first one in the sequence, then the pointer is null.

If the current PES stream is **aligned**, then each picture is encapsulated inside a single PES packet. It is of interest in this case, to associate each picture with the PES packet that encapsulates it. The field 7 in table 2 allows to associate the descriptor of the **aligned PES packet** with the descriptor of the picture it contains. Whether there is alignment or not is derivable from an information present in the PES Headers. In case of alignment, picture data along with the corresponding PES Header may be sent from memory to the PES parser without any further processing.

The content of the field 7 is set to null in case there is no alignment.

If the current stream is not aligned, then a picture may be spread over several PES packets. It is of interest in this case to identify the PES headers that start 'inside' a picture, either to remove them through processing in memory, or to modify them. In particular, the PES packet length item present in the PES Headers may need to be modified in case only partial PES packets are transmitted to the PES Parser. The field 8 in Table 2 points to the descriptor of the first PES packet contained within the data corresponding to the current picture. Each PES packet descriptor points itself to the descriptor of the following PES packet in the same picture.

The **SEU (Storage Elementary Unit) Address** is the address (Logical Block Address number), on the hard disk, of the SEU containing the picture header's first byte.

The **First Byte Address in SEU** is the offset in bytes, compared to the beginning of the SEU Address, of the first byte of the Picture header. It allows a direct access to the first byte of the picture. This information is derived by the Stream Parser.

The **Address of First Byte in PES Sequence** is the relative address between the first byte of the picture start code and the first byte of the whole video sequence that will be loaded into the memory for the edition during the restitution.

Items 10 and 12 enable loading of the SEUs containing all data relative to one picture. Thus, it is not necessary to load a whole sequence, and in order to reduce to amount of potentially unused data to transfer from the disk (i.e. avoiding the transfer of a whole sequence if only one picture is to be shown) only SEUs relative to a single picture may be transferred. It may be necessary to also load the corresponding sequence header to send the proper parameters to the video decoder.

The PES Header's contents may be required to properly decode and/or present the picture. Consequently, descriptors are also created for PES Headers.

Each PES descriptor ("E" descriptor) holds the following data items:

1	PES Index
2	First SEU Address
3	Address of the first byte in SEU
4	Number of SEUs containing the whole PES packet
5	Offset, in bytes, between the first byte of the PES packet and the first byte of the picture containing this PES packet
6	Pointer to the descriptor of the next PES in the same picture
7	Size of the PES packet in bytes
8	Size of the **PES packet's header** in bytes
9	Number of SEUs containing the whole **PES packet's header**

Table 3 – PES descriptor

The **PES index** gives the rank of a PES packet, compared to the beginning of the recorded video stream. It also plays the rule of an identifier of the PES descriptor.

The **SEU Address** is the number of the first LBA of the SEU (128 Kb block) containing the PES Header's first byte.

The **First Byte Address** in SEU is the offset in bytes, compared to the beginning of the SEU Address, of the first byte of the PES header.

Field 6 is a pointer to the descriptor of the next PES in the same picture. If there is no following PES packet in the same picture, then this pointer is null.

Given what has been said for tables 1 and 2, the other items of table 3 are self-explanatory.

Although this is not the case in the present embodiment, according to a variant embodiment, another descriptor is associated with Groups of Picture (GOP) and their headers.

The Temporal Index Table has the format shown by table 4.

Temporal Index	Sequence Descriptor Address	SEU Address
0	xxx	Yyy
1	xxx	Yyy
2	xxx	Yyy
3	xxx	Yyy
...	...	...
14400	xxx	yyy

Table 4 – Temporal Table

The temporal index corresponds to the number of seconds, counted from the beginning of a video stream. According to the present embodiment, 14400 entries are possible, corresponding to four hours of video, with one picture or frame representing 40ms.

The Sequence Descriptor Address gives the pointer to the Sequence descriptor containing the first picture after the temporal index, compared to the

corresponding VDU base address. If the corresponding VDU is not present in memory, it has to be loaded from the hard disk first, using information given in the VDU Table.

The SEU Address is the address, in LBA number on the hard disk drive, of the SEU containing

(a) in case of TS layer recording, the transport packet header of the transport stream packet containing the Sequence header of the first video sequence starting after T seconds,

(b) in case of PES layer recording, the sequence header of the first video sequence starting after T seconds.

For access to a video sequence starting approximately at a time T in seconds, it suffices to address the Temporal Table using T as an index and to use the corresponding SEU Address to start reading from the disk starting at the LBA which contains the beginning of the Transport Stream packet required to decode the sequence (in case (a)) or directly the sequence header location (in case (b)). Only the Temporal Table is required for such an access.

The rest of the data stored in both tables and VDUs is mainly used for Trickmode reproduction.

The VDU Table has the format shown by table 5:

VDU Index	**LBA Address**	**Size in LBAs**	**Temporal Table Index**
0	Xxx	yyy	Zzz to Qqq
1	Xxx	yyy	Zzz to Qqq
2	Xxx	yyy	Zzz to Qqq
...	...	...	...

Table 5 – VDU Table

The VDU Table has an entry for each VDU, and indicates for each VDU the number of the first LBA on the hard disk, the size of the VDU in terms of LBAs and the time interval (in seconds, starting from the beginning of the video stream) of the portion of video represented by the VDU. This interval specifies the entries into the TT Table.

The Temporal Index Table, the VDU Table and the VDUs are stored on the hard disk. The tables are also loaded into the trickmode buffer area 17 of memory 5, for modification in case of recording of video to the hard disk and for reference in case of reproduction from the hard disk. The necessary VDUs are read/written from/to the hard disk as required, depending also on the available quantity of free memory.

Generation of the trickmode data stored in the TT and VDU Tables and in the VDUs is carried out as follows.

The information to be generated is of three kinds: information extracted directly from the demultiplexed video packets, information describing the structure of the video stream and information relating to the location of certain video stream data on the hard disk drive. In the first case, a simple parsing of the PES or Picture headers in the stream yields the required information. In the second case, the video stream has to be analyzed and its structure memorized. In the third case, further information has to be sought from the file system. Table 6 indicates the origin of each kind of data.

Descriptor	Data	Origin
S	Sequence Index	Video stream analysis
S	PES Alignment	PES Header
S	SEQ & EXT Headers Size in bytes unit	Video stream analysis
S	SEQ & EXT Headers Size in SEU unit	Video stream analysis
S	Pointer to the Descriptor of First Picture	VDU structure
S	Pointer to the Descriptor of the Last Picture.	VDU structure
S	Pointer to the Descriptor of next Sequence	VDU structure
S	Pointer to the Descriptor of previous Sequence	VDU structure
S	Logical Block Address of the SEU containing the first byte of the Sequence	File System & Write FIFO management

S	Number of SEUs containing the whole sequence	File System & Write FIFO management
S	Offset, in bytes, of the first byte of the sequence inside the first SEU containing this byte.	File System & Write FIFO management & Video stream analysis
S	Size of the sequence in bytes	Video stream analysis
P	Picture Index	Video stream analysis
P	Type	Picture Header
P	Temporal Reference	Picture Header
P	Field/Frame	Picture Header
P	Pointer to the descriptor of the next picture in the same sequence	VDU structure
P	Pointer to the descriptor of the previous picture in the same sequence	VDU structure
P	Pointer to the descriptor of the **aligned** PES packet **containing** current picture.	VDU structure
P	Pointer to the descriptor of the first PES packet **starting inside** current picture.	VDU structure
P	SEU Address	File System & Write FIFO management
P	Number of SEUs containing the whole picture	File System & Write FIFO management
P	Offset, in bytes, of the first byte of the picture inside the first SEU containing this byte.	File System & Write FIFO management & Video stream analysis
P	Offset, in bytes, between the first byte of the picture and the first byte of the sequence containing this picture.	Video stream analysis
P	Size of the sequence in bytes	Video stream analysis

E	PES Index	Video Stream analysis
E	SEU Address	File System & Write FIFO management
E	Address of the first byte in SEU	File System & Write FIFO management & Video stream analysis
E	Number of SEUs containing the whole PES packet	File System & Write FIFO management & Video stream analysis
E	Offset, in bytes, between the first byte of the PES packet and the first byte of the picture containing this PES packet	Video Stream analysis
E	Pointer to the descriptor of the next PES in the same picture	VDU structure
E	Size of the PES packet in bytes	Video Stream analysis
E	Size of the **PES packet's header** in bytes	Video Stream analysis
E	Number of SEUs containing the whole **PES packet's header**	File System & Write FIFO management & Video stream analysis

Table 6

It is supposed in what follows that only one elementary video stream is recorded at a given time, i.e. only one Video PID is filtered. If more than one Video PID is filtered, the tables and VDUs are created in parallel and separately for each stream.

Parsing is carried out in a similar manner for TS and PES layer recording, i.e. the same items are spotted in the stored data. What changes is that for TS recording, when an item is spotted, the address of the TS header of the TS packet containing this item is used instead of the item's address.

The TS or PES packets stored by the demultiplexer in memory 8 are analyzed by first detecting Sequence headers, PES headers or Picture headers. Each of these headers has a predefined start code, defined by the MPEG II Video standard, and can easily be spotted in the incoming TS packet payloads or PES packets. Care has to be taken not to miss picture or sequence start codes spread over two PES packets, and picture, sequence or PES header

start codes spread over two TS packets. For each detected header, a corresponding descriptor (S, P, E) is created. PES and Picture headers are further parsed to extract the relevant fields to be inserted into the descriptors. Sequences, PES packets, and pictures are numbered starting from the first sequence, respectively PES packet or picture.

According to the present embodiment, a VDU holds only complete sequences. The maximum size of a VDU can be predefined, limiting the number of sequences stored into a VDU in accordance with the number of pictures per sequence and the presence of PES packets. Once the first sequence with all the associated pictures and PES packets has been described, the size of the memory space required to store the associated trickmode information is known and the number of sequences that will be described in each VDU can be determined.

Microprocessor 10 (through the File System) also determines the next SEU block address to which audio, video and/or other data is to be written. During the parsing process, the Stream Parser 6 determines the offset in bytes (or LBAs and bytes) of a given piece of data, compared to the beginning of the SEU. The offset is reset each time a SEU is written to the disk. Offsets are determined among other things for the following data items: for PES layer recording, Sequence headers, Picture headers, PES headers, and for TS layer recording, the addresses of the corresponding TS packet headers. SEU address and offsets for the three headers are inserted into the respective descriptors.

In parallel to the creation of the VDUs, the microprocessor creates the VDU Table and the Temporal Table.

An entry into the VDU Table is created every time a VDU is ready to be written to the disk. (According to the present embodiment, VDUs are written to a file of the stream file system). For each VDU, its position and size is given, in LBAs. The time interval it covers (in seconds, compared to the beginning of the video stream) is calculated, based on the number of pictures contained in the VDU. This information is also inserted into the VDU Table.

The Temporal Index Table comprises one entry per second according to the present embodiment. Its content is determined using the content of the VDUs and the TS header offsets (for TS layer recording) or the Sequence Header offsets (for PES layer recording).

Of course, depending on the specific application, the periodicity of the TT entries may be more or less than one second.

Both the VDU Table and the Temporal Index Table are written to the hard disk once they are created. Depending on their size and on the available memory, it may be required to split these tables and to load partial tables as required.

VDUs are intentionally made of linked elements using relative addressing to allow splitting and dynamic relocation in memory.

During the reproduction of the video, the analysis of the trickmode information is implemented using a cursor structure comprising two pointers : a pointer to sequence descriptors that leaps from a sequence to another, and a pointer to picture descriptors pointing to pictures inside the sequence pointed by the sequence pointer.

5. Trickmode restitution

According to the present embodiment, during trickmode video restitution, audio data is not transmitted to the audio decoder.

Reproduction from the hard disk drive for trickmode purposes will now be described. During this phase, the microprocessor 10 performs real-time stream edition of the previously recorded video stream, extraction and reordering of video access units (a unit being coded data relating to one picture) based on trickmode information, feeding of the decoder 9 and control of the decoding and display processes.

As the random access time to the hard disk drive is quite long, a realistic method is to read a slice of the recorded stream containing a single video sequence from the disk into the memory 5. The whole sequence being in memory 5, each picture (or other data such as a sequence header) in the sequence can be accessed to be transferred to the video decoder.

The PES parser 6 and/or the TS demultiplexer 7 remove the corresponding PES or TS layers, and extract information relevant to the lower layers from the PES headers, respectively TS headers. When receiving data, whether directly from the bus or from the demultiplexer 7, the PES parser will reject any data appearing before a valid PES header start code.

For trickmode reproduction, pictures in the stream are accessed one by one in memory, after a corresponding sequence has been read from the hard disk. However, whether in TS or PES recording mode, a PES header doesn't systematically directly precede the corresponding picture header. In other words, picture headers are not necessarily aligned on the beginning of a PES packet payload, and data irrelevant to the considered picture may exist between the PES header and the Picture header. For the PES parser to behave correctly, it is nevertheless necessary to supply this PES header. Else, the PES parser may not forward the picture data to the video decoder 9: all data preceding the first PES header is usually rejected after a decoder reset. Thus a Picture header followed by picture data not preceded by a PES header would also be rejected. According to the present embodiment, a dummy PES header is inserted before the Picture header of the picture to be decoded. A coherent PES stream is thus restored, with a minimum of irrelevant data being read from the hard disk and no irrelevant data being sent to the decoder 9 .

A simple example involving fast forward at twelve times the normal speed will be used to describe the insertion of the dummy PES header. For the purpose of this example, it will be supposed that only I-type pictures are accessed. Precautions to be taken when this is not the case, i.e. when the picture to be displayed is of the P or B-type, will be described later.

Fast forward at twelve times the normal speed involves reading and decoding one picture out of twelve (supposing only I-type pictures are accessed) and displaying the decoded pictures at the normal rate of one picture every 40 ms, in case of a 50 Hz frame rate.

(a) Stream edition at the PES layer level

The first task of microprocessor 10 is to determine the first video access unit to be extracted from the hard disk drive. Supposing that the fast forward starts at a time T compared to the beginning of the video stream, the first picture to be displayed is the first picture present in the stream after T.

In order to be used as an index in the VDU Table and the Temporal Table, T is truncated to an integer number of seconds. Using the VDU Table, the corresponding VDU is requested from the EIDE interface and loaded into memory (i.e. the trickmode buffer area), if it is not already present.

The Temporal Table points to the Sequence descriptor in this VDU containing the Picture descriptor. The Sequence descriptor's contents are used

to load the corresponding whole video sequence into memory 5. Decoder 9 is such that the microprocessor 10 can adjust the decoding parameters of the decoder 9. It may then not be necessary to transmit Sequence Headers to the PES Parser before transmitting the dummy PES Headers followed by the Picture data.

Each picture representing 40 ms and using the Picture List (which points to the different Picture descriptors of the pictures in the sequence), it is easy to access the Picture descriptor which in time is the closest to T. The Picture descriptor indicates the offset of the picture header in the video sequence loaded in memory. Thus the desired picture is sent to the decoder and the decoder is programmed by microprocessor 10 to correctly handle this picture.

In this case, data is provided from memory 5 to the PES Parser 6, since the transport layer has already been removed.

Figures 7a and 7b represent a PES stream under the form of sequential pictures mapped into PES packets containing a picture to be decoded. The represented part of the stream may be that stored in the video read FIFO, assuming that only the PES layer was recorded. Each picture data is preceded by a Picture header, both together forming a video access unit. The stream includes PES headers at positions generally independent of the content of the elementary video stream.

Figure 7a shows the unedited PES stream, picture n being the picture to be displayed. It is preceded by a header. The header of the PES packet containing the Picture header of picture n is indicated by an arrow. Figure 7b shows the edited PES stream, into which the microprocessor 10 has inserted the dummy PES header, just in front of the Picture header of picture n, so as to avoid any intervening data between the two headers.

According to the present embodiment, the dummy PES header has the format given in table 7. It is the shortest header allowed by the MPEG II Systems document (i.e. 9 bytes), and is sent to the video decoder 9 before the content of the video read FIFO is read starting from the address defined by the Picture header offset. The decoder will then see a valid PES stream and process the picture data as instructed by the microprocessor 10.

A dummy PES header is inserted each time there is a gap in the sequence of video access units to be sent to the decoder.

In the tables below, the notation '0x' designates hexadecimal values. The lower case letter 'u' designates a variable binary value.

Value	Signification
0000 0000 0000 0000 0000 0001 ("0x000001")	Packet start code prefix
1110 uuuu (where 'uuuu' varies between "0xE0 -> 0xEF")	Video Stream number $(uuuu)_b$
0000 0000 0000 0000 ("0x0000")	PES packet length
$10_{(7:6)}$ $uu_{(5:4)}$ $0_{(3)}$ $0_{(2)}$ $0_{(1)}$ $0_{(0)}$	***PES_scrambling_control*** These two bits shall be a copy of the same bits from the previous PES packet header **PES_Priority** low ***Data_alignment_indicator:*** It is not defined whether a video start code immediately follows the PES Header or not ***Copyright*** ***Original_or_copy***: PES packet payload is a copy
$00_{(7:6)}$	***PTS_DTA_Flag:*** No PTS and DTS fields present
$0_{(5)}$	***ESCR_Flag:*** No ESCR fields present
$0_{(4)}$	***ES_rate_Flag***: No ES_rate field is present
$0_{(3)}$	***DSM_trick_mode_Flag:*** No DSM_trick_mode field is present
$0_{(2)}$	***Additional_copy_info_Flag:*** Corresponding field is not present

$0_{(1)}$	***PES_CRC_Flag:*** Corresponding field is not present
$0_{(0)}$	***PES_extension_Flag:*** Corresponding field is not present
0000 0000 ("0x00")	PES packet header length (no more bytes in the PES header)

Table 7 – Dummy PES Header

(b) Stream edition at the TS layer level

In this case, data is transferred from the read FIFO 22 to the Transport Stream demultiplexer 7.

The TS layer has an additional constraint compared to the PES layer: editing can only be done at the TS packet level, i.e. a whole TS packet has to be added or removed. Inserting or deleting bytes in existing packets results in an invalid TS stream since a TS packet has a fixed length of 188 bytes.

Consequently, determination of the SEU containing the picture to be decoded is carried out in a slightly different way compared to case (a). Again, a whole slice of stream containing the video sequence containing the picture is loaded in memory 5. In order to comply with the requirement to submit only entire TS packets to the demultiplexer 7, it is required to start reading starting from the TS header of the TS packet containing the Picture header of the picture to be decoded. The TrickModes information provides for the necessary address information : in case of TS stream recording, all addresses in TrickModes information descriptors are suitably aligned on the TS packet borders.

Figure 8a represents a TS stream of packets of the same video component (i.e. having same PID) containing the Picture header of the picture to be decoded.

Instead of inserting only a PES header, a whole TS packet is inserted. This TS packet also contains a dummy PES header, for the same reasons as for (a). Figure 8b illustrates the stream after the TS packet insertion.

The inserted TS packet header contains the same PID value as that of the TS packet header of the TS packet comprising the Picture header. The TS packet header also contains a continuity_count value which is equal to that of the TS packet header of the TS packet containing the Picture header, decremented by one and taken modulo 16, to be consistent with the following TS packet's value. The continuity count value is directly read in the stream in

memory. Among the adaptation field flags, the discontinuity error flag is set to indicate a discontinuity compared to any previous continuity_count value. The adaptation field's length is chosen so that the length of the entire TS packet, header included, is 188 bytes.

The TS payload contains the dummy PES header, already described. Contrary to case (a), irrelevant or unusable data may be present between the PES header and the Picture header of the picture to be decoded, since the Picture header is not necessarily aligned with the end of the TS header. In order to inform the video decoder to ignore this irrelevant data, the inserted TS header also contains a sequence error code, after the dummy PES header. Figure 9 illustrates the data received by the decoder, PES layer removed. Picture X is the picture to be decoded. The decoder's input buffer still contains partial data previously received, concerning a picture B+1, resulting from the transfer of a previous picture to be decoded, for instance picture B. Data relating to picture X-1 is the data present between the dummy TS packet inserted by microprocessor 10 and the Picture Header of picture X. The TS header of the dummy TS packet has been removed by the demultiplexer 7, and the dummy PES header contained in the dummy TS packet's payload has been removed by the PES Parser 6. Between the partial data of pictures B+1 and X-1, there remains the error sequence code ("0x00 00 01 B4") preceded by another code ("0xB4").

Upon detection of the sequence error code, mentioned among others in Section 6.2.1. and Table 6-1 of the MPEG II Video document, the decoder 9 rejects all data received before the error code, and all data received in the future, up to the next Picture Header. Decoder 9 is constructed so as to have this behaviour.

A new problem is introduced by the insertion of the sequence error code: once the PES parser 17 has rid the stream of the PES headers, it may happen that the last bytes of the payload of the PES packet preceding the inserted PES packet, combined with the first byte of the sequence error code (i.e. "0x00") constitute a Picture header start code (i.e. "0x00 00 01 00"). To avoid this case, a byte of value "0xB4" is inserted between the dummy PES header and the sequence error code. In this case, if the last three bytes of the preceding PES packet payload are indeed "0x00 00 01", then the formed code is another sequence error code "0x00 00 01 B4". Whether this code is present once or twice is not important as far as the behavior of the video decoder is

concerned. When the last three bytes and the "0xB4" byte do not form the sequence error code, the presence of the B4 is of no consequence because the following sequence error code will in any event eliminate the previous contents of the video decoder's input buffer, including the additional "0xB4".

Value	Signification
HEADER	
0x47	Sync Byte
$0_{(7)}$ $1_{(6)}$ $0_{(5)}$ (PID 5 MSB) $_{(4:0)}$	No error in the TS packet Start of a PES packet Transport priority low 5 MSB of the video stream component PID
(PID 8 LSB) $_{(7:0)}$	8 LSB of the video stream component PID
$00_{(7:6)}$ $11_{(5:4)}$ $(N-1)_{(3:0)}$	TS payload not scrambled Adaptation field followed by payload ***continuity_count*** : takes the value of the following TS packet continuity_count minus 1 (modulo 16)
0xA9 (169 $_{Dec}$)	Adaptation field length: Equals to 183 minus the sum of : ***- dummy PES header*** length (9 bytes) ***- sequence error code*** length (5 bytes)
$1_{(7)}$ $0000000_{(6:0)}$	Adaptation field flags: Discontinuity error is set.
0xFF * 168	168 stuffing bytes
PAYLOAD	
0x00 0x00 0x01 (1110 uuuu)b 0x00 0x00 (10uu 0000)b 0x00 0x00	***dummy PES header***

0xB4	Ensures that the code given hereafter will not be included into an undesired sequence.
00 00 01 B4	***sequence error code***

Table 8 – TS packet with dummy PES Header and sequence error code

Normally, pictures accessed one by one during trickmode are not necessarily intra-type pictures. It may thus be necessary to decode other pictures and to maintain them in memory in order to decode a particular picture. If the picture to be displayed is of the P-type picture, then it will be necessary to decode the preceding I-type picture (which can be found using the Picture descriptors preceding the Picture descriptor of the picture to be displayed) and to decode that I-type picture first. It must be remembered that pictures are transmitted – and stored – according to the order in which they are to be decoded, not the order according to which they are to be displayed. This order generally differs from the displaying order. The video decoder will be instructed by the microprocessor 6 to only decode the I-type picture, but not to display it. The P-type picture is then decoded and displayed.

Similarly, if a B-type picture is to be decoded, the preceding and following I and/or P type pictures have to be extracted from the hard disk and decoded first.

The present embodiment concerns mostly TS stream packet recording and reproduction, but of course the recording/reproduction of other layers, in particular the PES layer, is not outside of the scope of the invention.

Moreover, although according to the present embodiment, the microprocessor 6 manages the file systems of the hard disk drive, this task may also be performed by another processor in the receiver, in particular the video decoder 10.

Also, although the mass storage device used in the present embodiment is a hard disk drive, another type of device could also be used. For example, recordable Compact Discs or Digital Video Discs may be employed.

Claims

1. Method for generating trickmode information for a digital video stream to be recorded in a digital video system, characterized by the steps of:

analyzing the structure of said video stream and deriving a plurality of descriptors of objects of the stream;

writing said stream to a recording medium;

inserting, into the object descriptors, information describing the address of the objects on the recording medium.

2. Method according to claim 1, wherein each descriptor contains

information identifying recording medium data blocks containing the object associated with the descriptor;

information identifying the location of the object within said recording medium data blocks.

3. Method according to claim 1 or 2, wherein each descriptor contains a link to the preceding descriptor and/or to the following descriptor.

4. Method according to one of the claims 1 to 3, wherein each descriptor contains an identifier giving the rank of the object relative to the beginning of the recorded stream.

5. Method according to one of the claims 1 to 4, further comprising the steps of:

assembling descriptors of objects corresponding to a time interval within a video description unit,

generating a video description unit indexing table of all video description units.

6. Method according to claim 5, wherein a video description unit comprises all descriptors relating to N sequences of pictures, where N is an integer greater or equal to 1.

7. Method according to one of the claims 5 or 6, wherein pointers to descriptors are given relative to a video description unit base address.

8. Method according to one of the claims 1 to 7, wherein the descriptors include descriptors relating to picture sequences, pictures and PES packets.

9. Method according to claim 8, wherein a sequence descriptor comprises a pointer to the first and the last picture descriptors in the sequence.

10. Method according to one of the claims 8 or 9, wherein a sequence descriptor contains data identifying the location of the sequence header and its extensions in the recording medium data blocks.

11. Method according to one of the claims 8 to 10, wherein a picture descriptor contains an information on whether PES packets and picture headers are aligned or not.

12. Method according to claim 3 and one of the claims 8 to 10, wherein a picture descriptor contains a pointer to a PES descriptor corresponding to a PES header inserted within the recorded picture data.

13. Method according to one of the claims 8 to 12, wherein a picture descriptor further comprises a pointer to the sequence descriptor of the sequence containing the picture.

14. Method according to claim 5 combined with any one of the claims 1 to 4 or 6 to 13, further comprising the step of recording the video description units on the recording medium and inserting addresses of the video description units on the recording medium into the video description unit table.

15. Method according to claim 5 combined with any one of the claims 1 to 4 and 6 to 14, further comprising the steps of including into the video description unit table, for each video description unit, the following data:

- the address of the first recording medium data block in which the video description unit is recorded,
- the number of recording medium data blocks occupied by the video description unit.

16. Method according to claim 5 in combination with any of the claims 1 to 4 and 6 to 15, wherein the video description unit table further comprises, for each video description unit, an information defining the time interval of the video stream described by the unit.

17. Method according to one of the claims 1 to 16, further comprising the step of generating a temporal index table containing a plurality of time indexes, and for each index, an address on the recording medium of the beginning of the data necessary for restituting the video stream starting from said index.

18. Method according to claim 5 combined with claim 17, wherein said temporal index table further comprises, for each index, a pointer, relative to a base address of a video description unit, of the sequence descriptor corresponding to a given index.

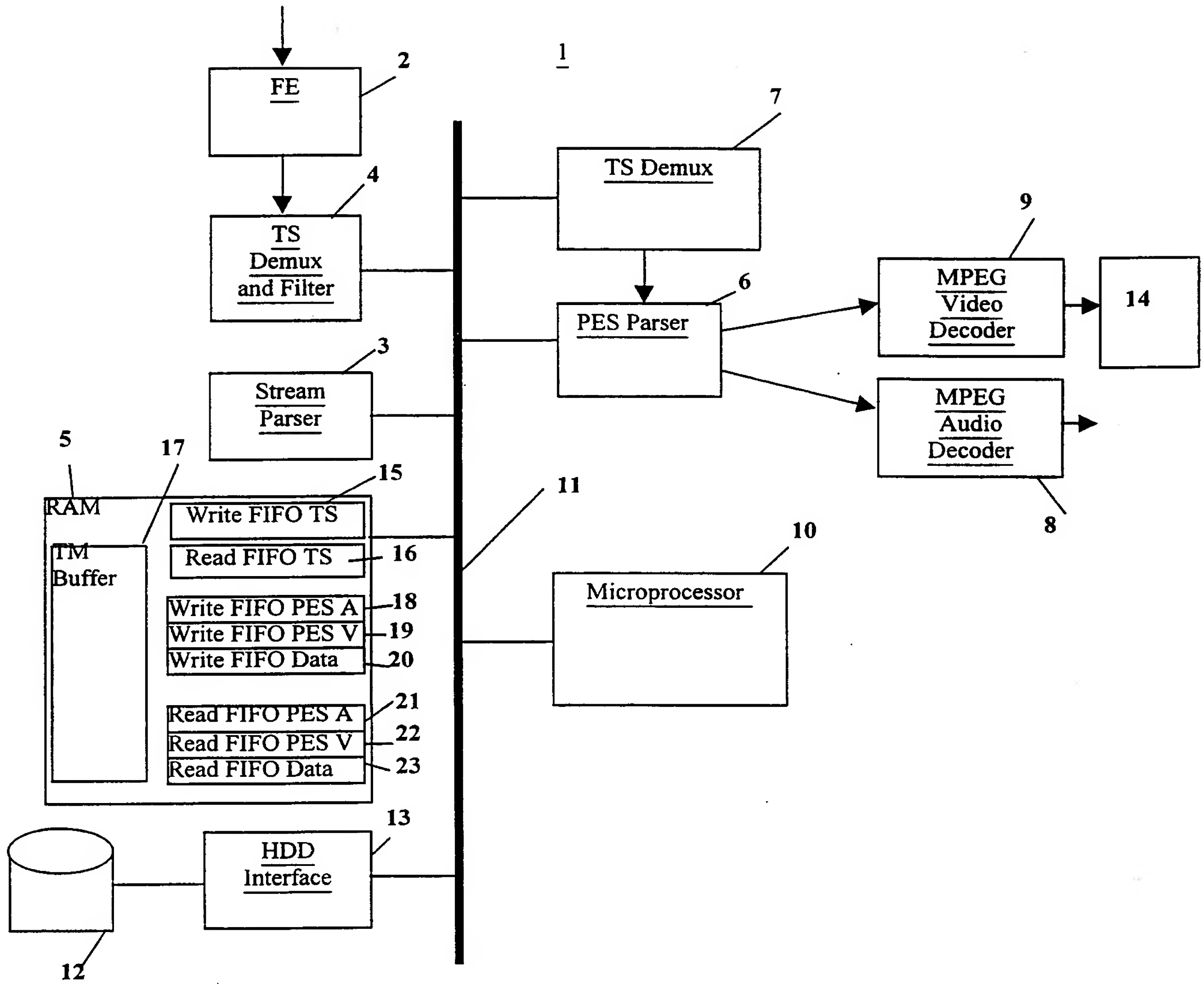


Fig. 1 : System Overview


Boot Block
File System Data
'Streamed' Audio/Video/Data

Fig. 2


Super block
Nodes
Run Extensions
Bit Tables
Audio/Video Data

Fig. 3

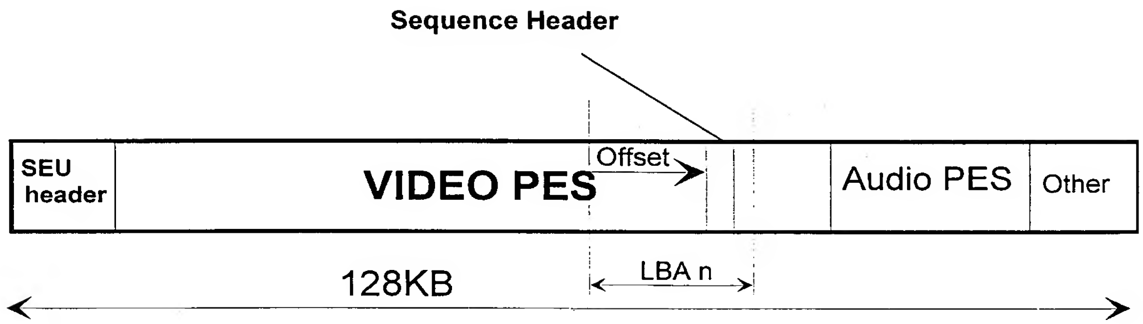

Sequence Header
SEU header
VIDEO PES
Offset
Audio PES
Other
128KB
LBA n

Fig. 4a

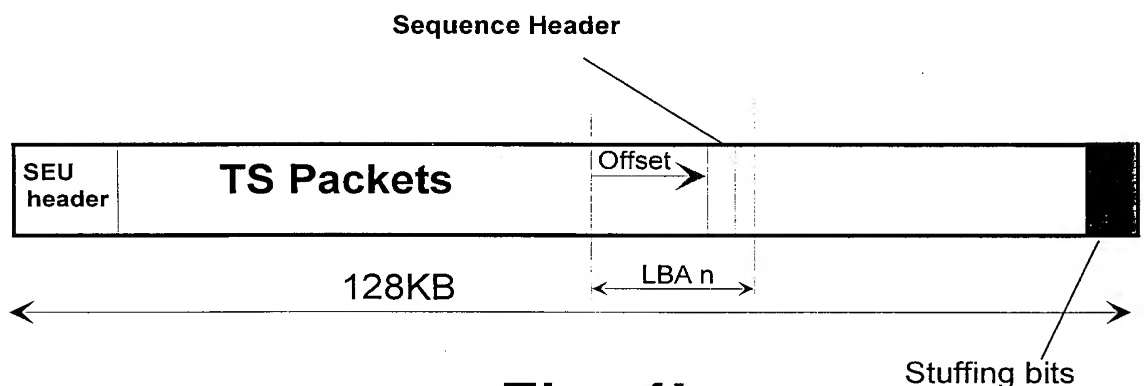
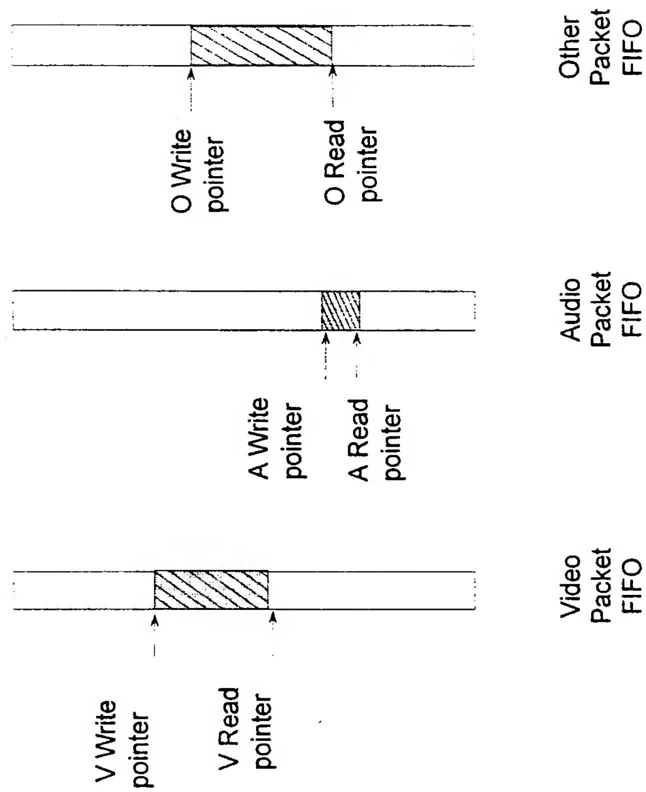
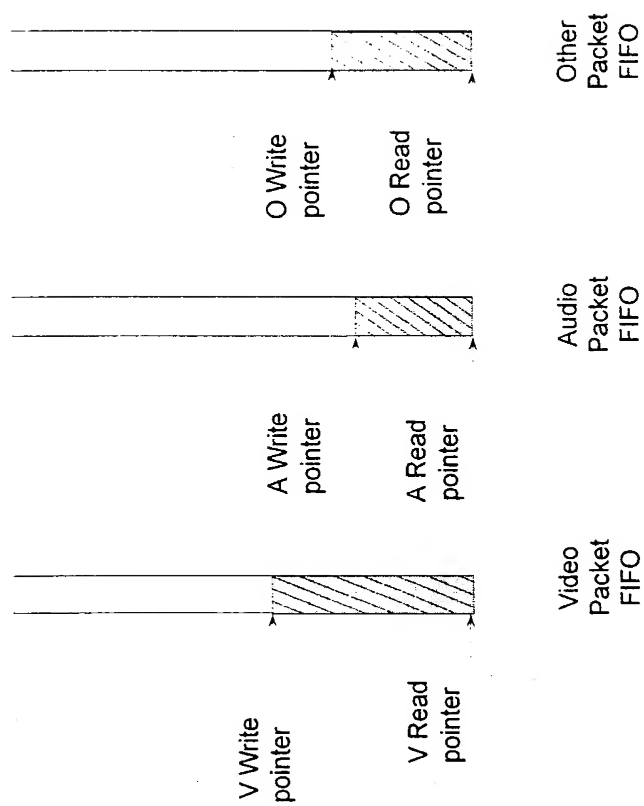

Sequence Header
SEU header
TS Packets
Offset
128KB
LBA n
Stuffing bits

Fig. 4b



V Write pointer
V Read pointer
Video Packet FIFO
A Write pointer
A Read pointer
Audio Packet FIFO
O Write pointer
O Read pointer
Other Packet FIFO

Fig. 5b



V Write pointer
V Read pointer
Video Packet FIFO
A Write pointer
A Read pointer
Audio Packet FIFO
O Write pointer
O Read pointer
Other Packet FIFO

Fig. 5a

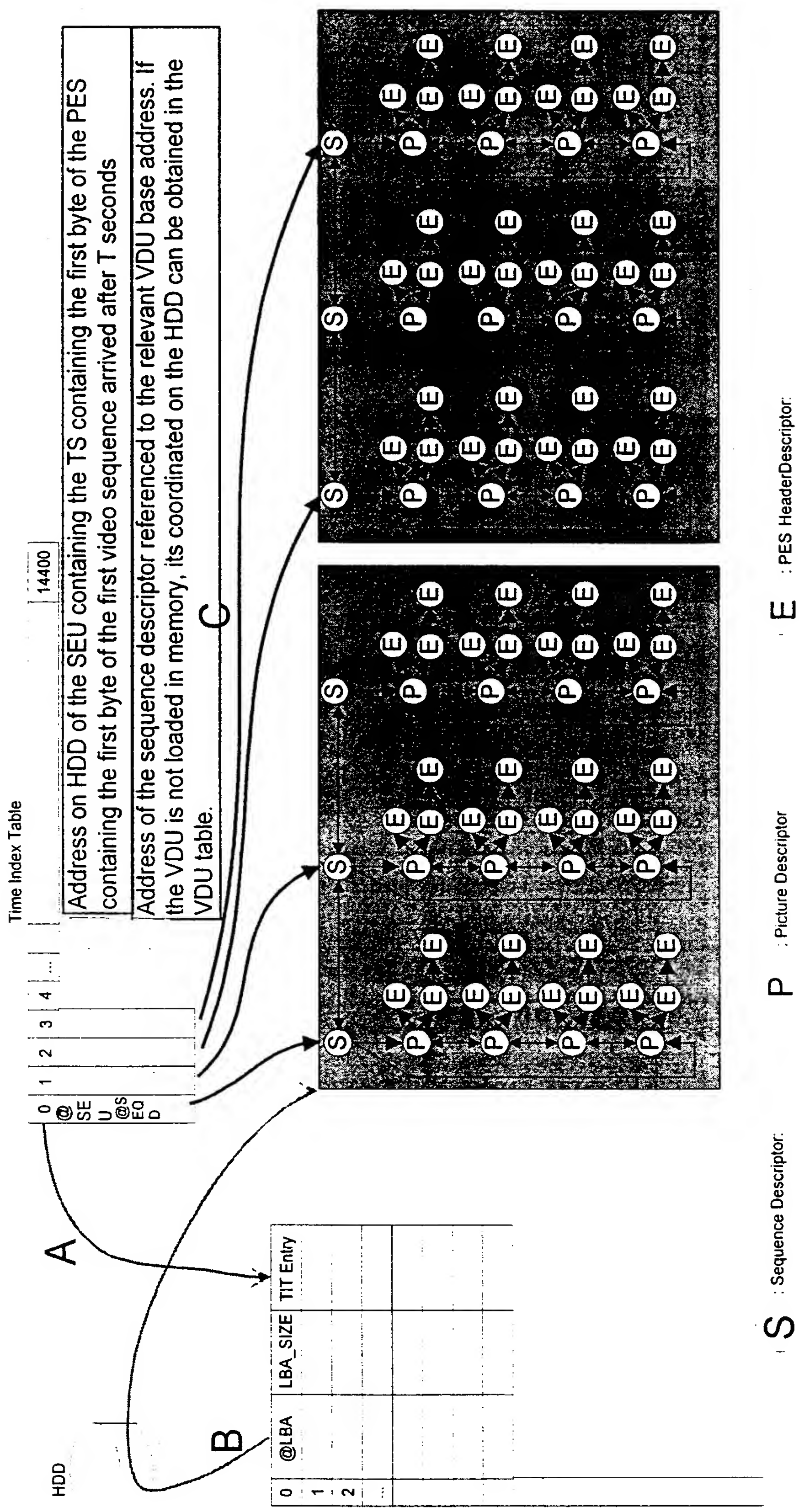

Time Index Table
0
1
2
3
4
...
14400
@SEU
@SEQD
Address on HDD of the SEU containing the TS containing the first byte of the PES containing the first byte of the first video sequence arrived after T seconds
Address of the sequence descriptor referenced to the relevant VDU base address. If the VDU is not loaded in memory, its coordinated on the HDD can be obtained in the VDU table.
A
B
C
HDD
@LBA
LBA_SIZE
TIT Entry
S
P
E
S : Sequence Descriptor
P : Picture Descriptor
E : PES HeaderDescriptor

Fig. 6

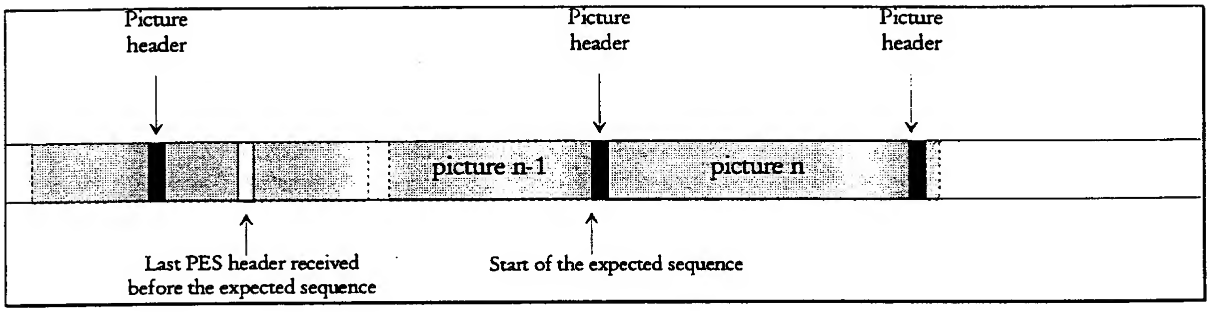

Picture header
Picture header
Picture header
picture n-1
picture n
Last PES header received before the expected sequence
Start of the expected sequence

Fig. 7a

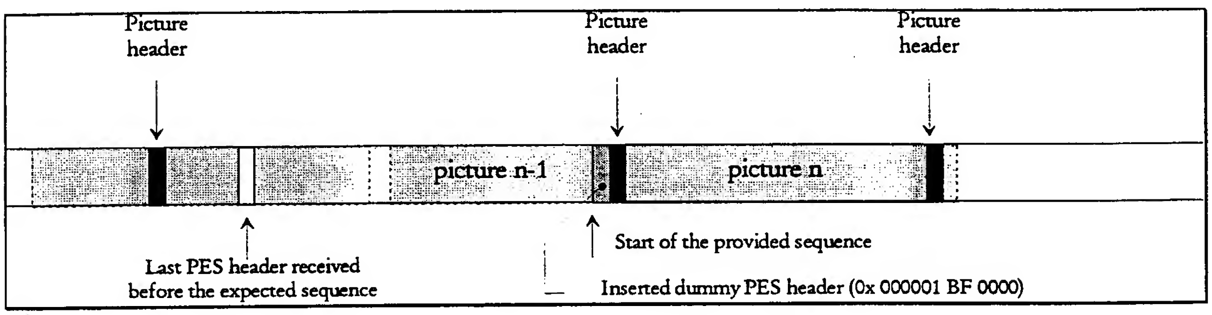

Picture header
Picture header
Picture header
picture n-1
picture n
Start of the provided sequence
Last PES header received before the expected sequence
Inserted dummy PES header (0x 000001 BF 0000)

Fig. 7b

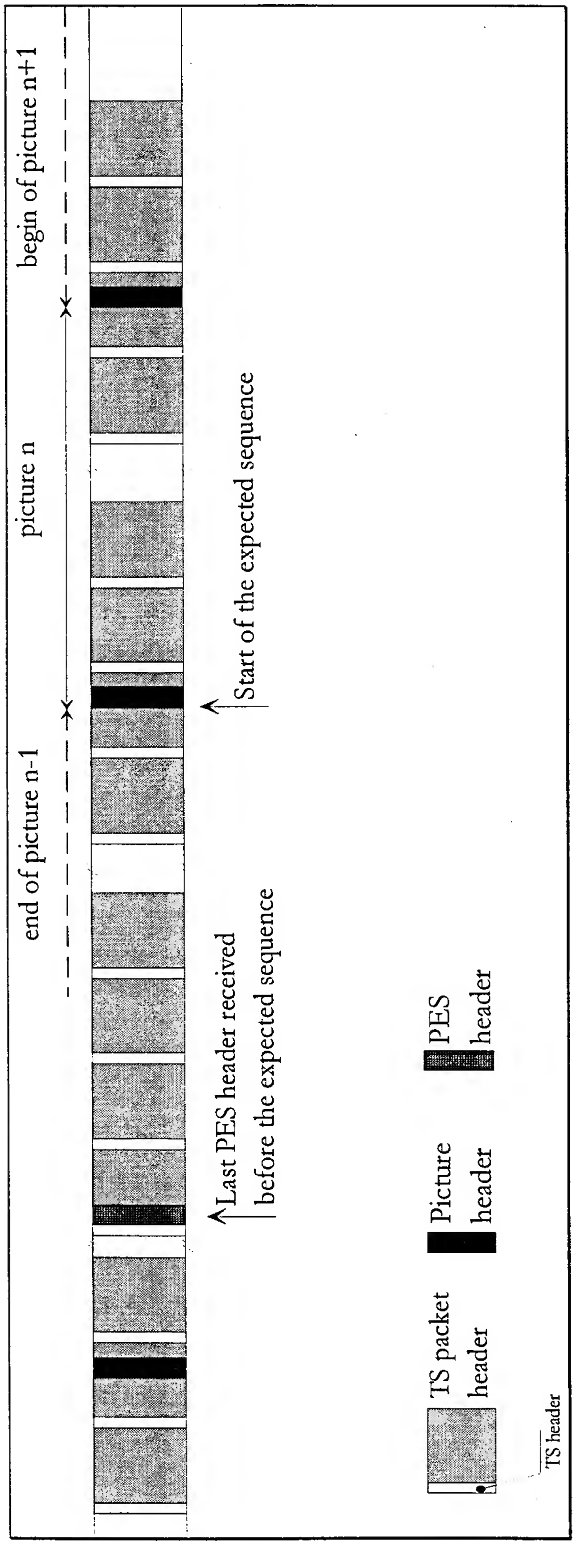

begin of picture n+1
picture n
end of picture n-1
Start of the expected sequence
Last PES header received
before the expected sequence
TS packet
header
Picture
header
PES
header
TS header

Fig. 8a

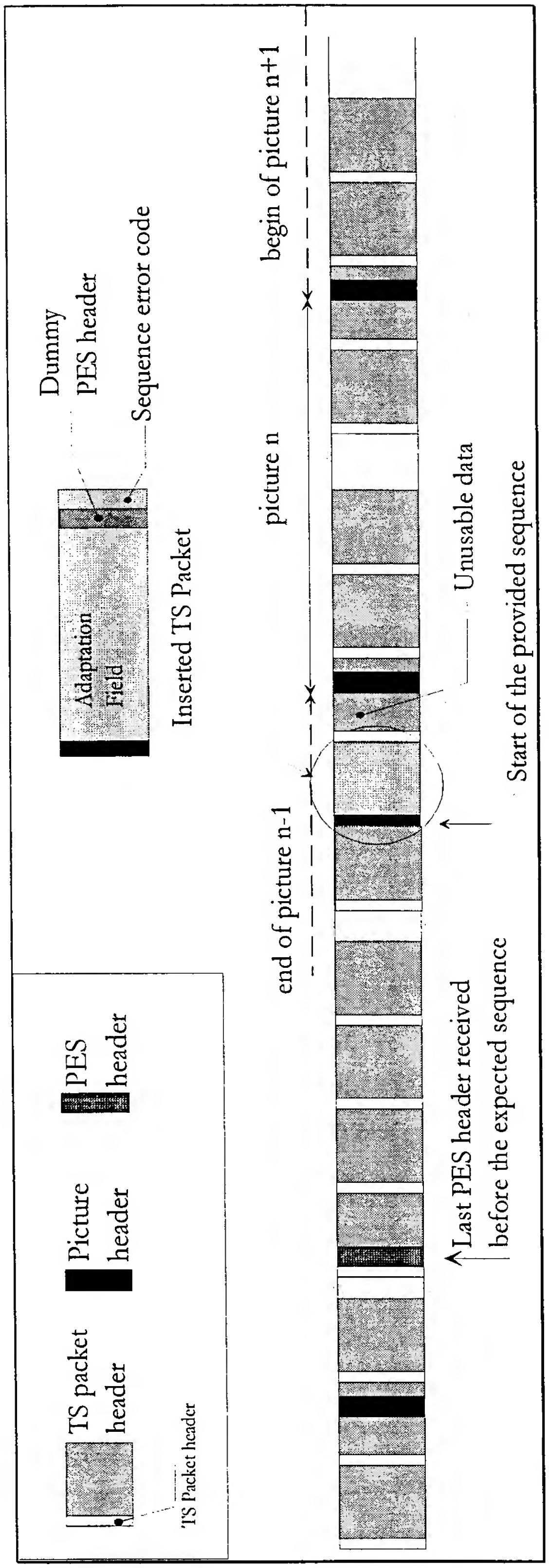

Dummy
PES header
Sequence error code
Adaptation Field
Inserted TS Packet
begin of picture n+1
picture n
end of picture n-1
Unusable data
Start of the provided sequence
Last PES header received
before the expected sequence
TS packet header
Picture header
PES header
TS Packet header

Fig. 8b

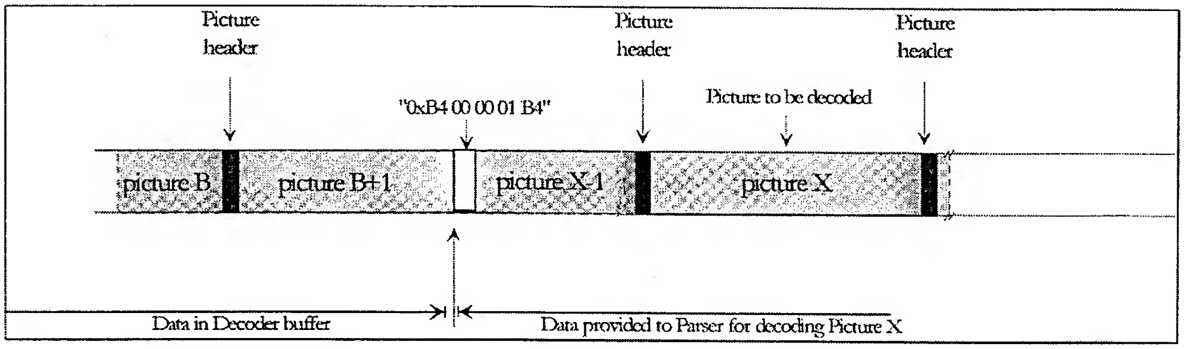

Picture header
Picture header
Picture header
Picture to be decoded
"0xB4 00 00 01 B4"
picture B
picture B+1
picture X-1
picture X
Data in Decoder buffer
Data provided to Parser for decoding Picture X

Fig. 9

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP 00/09919

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/24 H04N5/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N G11B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 695 098 A (SHARP KK) 31 January 1996 (1996-01-31)	1-7,14
A	column 8, line 32 -column 9, line 21 column 9, line 42 -column 11, line 39 column 12, line 23 -column 15, line 44	8-18
X	EP 0 729 153 A (HITACHI LTD) 28 August 1996 (1996-08-28) column 3, line 35 -column 4, line 41	1,2,4
X	US 5 481 543 A (VELTMAN MARKUS H) 2 January 1996 (1996-01-02)	1,2,8
A	column 22, line 26 - line 61 column 23, line 14 - line 32 column 28, line 21 - line 51	4

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 February 2001	06/03/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Marie-Julie, J-M

3

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte .ional Application No
PCT/EP 00/09919

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0695098 A	31-01-1996	JP 8045249 A	16-02-1996
		DE 69519607 D	18-01-2001
		EP 1028426 A	16-08-2000
		US 6021250 A	01-02-2000
EP 0729153 A	28-08-1996	JP 8235833 A	13-09-1996
		CN 1135072 A	06-11-1996
		CN 1229235 A	22-09-1999
		EP 0930618 A	21-07-1999
		KR 227417 B	01-11-1999
		US 6002834 A	14-12-1999
		US 6009237 A	28-12-1999
US 5481543 A	02-01-1996	AU 682045 B	18-09-1997
		AU 6936494 A	03-01-1995
		CA 2141672 A	22-12-1994
		CN 1110893 A	25-10-1995
		DE 69418678 D	01-07-1999
		DE 69418678 T	21-10-1999
		EP 0654199 A	24-05-1995
		ES 2132406 T	16-08-1999
		WO 9430014 A	22-12-1994
		JP 8505024 T	28-05-1996
		KR 205494 B	01-07-1999
		AU 672070 B	19-09-1996
		AU 5483494 A	01-09-1994
		DE 69417795 D	20-05-1999
		DE 69417795 T	12-08-1999
		EP 0618728 A	05-10-1994
		JP 6343065 A	13-12-1994